

# Optimizing Compiler for Engineering Problems

Petr R. Ivankov

August 22, 2008

## Abstract

New information technologies provide a lot of prospects for performance improvement. One of them is "Dynamic Source Code Generation and Compilation". This article shows how this way provides high performance for engineering problems.

## 1 Introduction

Present day engineering problems require multidisciplinary usage of computer. For example engineers use following IT branches: math calculations, 3D graphics, real time control, databases. Software devoted for every single branch is not effective. This circumstance reflects common integration strategy of IT. There exists a myth that performance requires specialization and contradicts universality and/or integration strategy. This article shows that very universal software can provide very high performance. Section 2 is devoted to universal software. Section 3 describes runtime optimizing for it.

## 2 Universal Framework for Science and Engineering

Information about universal engineering software is contained at following web cites.

- Universal Framework for Science and Engineering homepage  
<http://www.mathframe.com/>;
- CodeProject articles devoted to framework applications  
[http://www.codeproject.com/script/articles/list\\_articles.asp?userid=3080377](http://www.codeproject.com/script/articles/list_articles.asp?userid=3080377);
- Arxiv articles [1], [2] devoted to framework applications.
- SourceForge projects  
AstroFrame <https://sourceforge.net/projects/astrohalaxy>;  
CategoryTheory <https://sourceforge.net/projects/categorytheory/>;

Described framework is based on three principles. First one is component approach. Second principle is insertion of math formulas. Third principle is openness of framework. So let us consider them.

## 2.1 Component approach

The best method of complicated phenomenon grasping is a decomposition of the phenomenon. The phenomenon contains objects (components) and there exist links between objects. Any object may belong to a set of domains. For example a source of physical field [8] has a geometrical position. Hence it is a subject of positioning domain. This object may be linked to other object of positioning domain by positioning links. If source of physical field receives and then transmits information then it is a consumer of information. So it is also a subject of information domain and may be linked to sources of information. And at last it is a subject of physical field domain. Typical picture of objects (components) of virtual reality simulation is presented on Figure 1. This picture

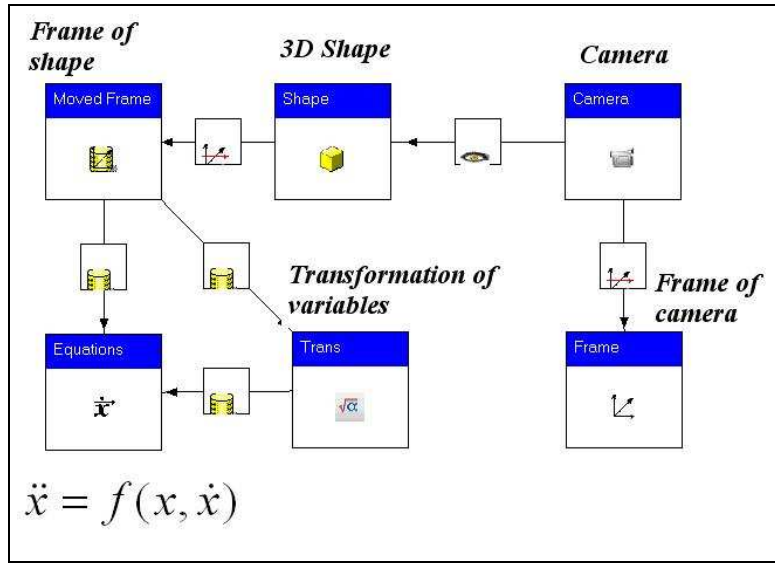


Figure 1: Typical example of virtual reality

shows motion of 3D shape. We have ordinary differential equations of the shape motion

$$\ddot{x} = f(x, \dot{x}).$$

The component at left bottom is a solver of these equations. It is a source of information. In this situation we should perform some transformation of this information. To do this we use “Transformation of variables”. Later component

is a consumer of information of the solver. So it is connected to solver by information link. We also have a “Frame of shape”. It is a moved reference frame that uses information from the solver and from the “Transformation of variables”. So it is a consumer of information and it is connected to its providers of information. The “3D Shape” is rigidly connected to this frame. The shape is connected to the frame by geometrical positioning link. Also we have a virtual camera and its reference frame. They are also connected by geometrical positioning link. And at last the camera is connected to the shape by visibility link.

## 2.2 Formula editor

We need rich formula editor for interdisciplinary software. Insertion of formulas enables us use them in different tasks of virtual reality. These formulas may be used for definition of signals, transformation of 3D figures, definition of right parts of differential equations, and definition of size and color of figures, etc. The formulas may contain real, integer, boolean, vector variables etc. In the CategoryTheory project <https://sourceforge.net/projects/categorytheory/> formula, the editor operates even with Galois fields. Implemented in related projects formula editor formula editor is case sensitive and operates with a lots of types. Result of calculation depends on types of variables. Let  $\sin(a)$  be a formula of formula editor. What is its meaning? If  $a$  is a real variable then result of formula is a real value. However if  $a$  is an array then  $\sin(a)$  is also an array of componentwise calculation of  $\sin$ . If  $a$  is a real array and  $b$  is a real variable then  $a + b$  means an array of sums of components of  $a$  with  $b$ . Any function of formula editor may be a variable or result of calculation. A function as result is not a value of function but it is the whole function. This fact seems unusual for people who do not know functional analysis. For example if  $a$  is a real variable then  $f(a)$  means result of calculation of  $f$ . If  $a$  is a function then  $f(a)$  is a composition of functions. Formula editor supports matrix and vector operations. Examples of usage of vector and matrix operations are presented below:

$$\begin{aligned} f^t a f, \\ (q^{-1} + h)^{-1}, \\ a \times b. \end{aligned}$$

These examples contains transposition of matrixes, products of matrixes, inversion of matrixes and vector product of 3D vectors. A very good sample of these operations' applications is Kalman filter [9]. In particular this filter is used in motion control systems. You can download and evaluate example of this filter from: <http://www.codeproject.com/cs/library/UniversalEnggFrmwork6.asp> The formula editor supports Dirac delta function. The presence of the delta function at the right part of the ordinary differential equation shows that the result function is not continuous. Following picture shows presence of delta function in formula editor

$$f(t)\delta(t)$$

## 2.3 Openness of the software

Usually, every developer or company has its own projects of those object domain. This software does not require discarding of existing projects. Any object domain project may be included to this software. If you wish to include your project or its part, then you should develop an adapter, compile the class library, and link it to this software. The adapter should contain one or more classes that implement one or more interfaces of this software. A more profound description of these interfaces is contained in the developer's guide. You can download the guide from AstroFrame homepage <https://sourceforge.net/projects/astrohalaxy>.

## 3 Optimizing Compiler

Optimizing compiler transforms formula editor expressions to C# code. Then CodeDom

<http://msdn.microsoft.com/en-us/library/650ax5cx.aspx>

performs compilation. Generated code do not contain cycles or unnecessary assignments. Let us consider some examples of generated code.

### 3.1 Matrix Product

Formula editor indicates matrix product as  $ab$  where  $a$  and  $b$  are first and second factor respectively. Dimension of  $a$  and  $b$  depends on framework project. Generated code looks like:

```
var_2[0, 0] = var_0[0, 0] * var_1[0, 0] + var_0[0, 1] * var_1[1, 0] +
var_0[0, 2] * var_1[2, 0] + var_0[0, 3] * var_1[3, 0];
var_2[0, 1] = var_0[0, 0] * var_1[0, 1] + var_0[0, 1] * var_1[1, 1] +
var_0[0, 2] * var_1[2, 1] + var_0[0, 3] * var_1[3, 1];
...
```

where `var_0`, `var_1` and `var_2` corresponds to  $a$ ,  $b$  and  $ab$  respectively. So we have code without cycles and unnecessary assignments. It is possible code generation even without arrays.

### 3.2 Vector Calculation

Formula editor is context sensitive. It has high level of abstraction. For example if  $a$  and  $b$  are vectors then  $a+b$  is vector sum. But if  $a$  is a scalar then  $a+b$  means componentwise sum to scalar. Following code shows this situation: `var_2[0] = var_0 + var_1[0];`

```
var_2[1] = var_0 + var_1[1];
var_2[2] = var_0 + var_1[2];
...
```

where `var_0`, `var_1` and `var_2` corresponds to  $a$ ,  $b$  and result of calculation respectively.

### 3.3 Sum

If  $a$  is number array then formula editor can represent sum of its elements by the following way

$$\sum a; \quad (1)$$

`var_1 = var_0[0] + var_0[1] + var_0[2] + var_0[3] + ...;`

where `var_0`, `var_1` corresponds to  $a$  and  $\sum a$  respectively. Generated code is presented below:

### 3.4 Symbolic Differentiation

Framework contains symbolic differentiation. Expression  $\frac{d}{dt} \sin t^2$  match to the following code.

`var_1 = 2 * var_0 * Math.Sin(var_0);`

where `var_0`, `var_1` corresponds to  $t$  and result of differentiation.

### 3.5 Conclusion

There exists opinion that .NET does not match to scientific and engineering problems. One reason that .NET requires garbage collection. However the framework construct new objects during project loading only. Main calculations do not contain construction and therefore they do not require garbage collection. Second reason is that .NET has slow performance. This article shows that runtime compilation enables us reach very high performance. Besides this circumstances .NET have a lot of other properties that are useful for scientific and engineering problems. Thank you very much.

## References

- [1] Petr R. Ivankov, Nikolay P. Ivankov. The framework for simulation of dynamics of mechanical aggregates. arXiv:cs/0701119.
- [2] Petr R. Ivankov, Nikolay P. Ivankov. The virtual reality framework for engineering objects. arXiv:cs/0612126.
- [3] A new approach to magnetic angular momentum management for large scientific satellites. Iida H. (1), Ninomya K. <http://cat.inist.fr/?aModele=afficheN&cpsidt=3049179>
- [4] Spin-stabilisation, Wikipedia <http://en.wikipedia.org/wiki/Spin-stabilisation>
- [5] Elasticity (physics), Wikipedia [http://en.wikipedia.org/wiki/Elasticity\\_%28physics%29](http://en.wikipedia.org/wiki/Elasticity_%28physics%29)
- [6] Finite field, Wikipedia [http://en.wikipedia.org/wiki/Galois\\_field](http://en.wikipedia.org/wiki/Galois_field)
- [7] Category theory, Wikipedia [http://en.wikipedia.org/wiki/Category\\_Theory](http://en.wikipedia.org/wiki/Category_Theory)
- [8] Field (physics), Wikipedia [http://en.wikipedia.org/wiki/Field\\_%28physics%29](http://en.wikipedia.org/wiki/Field_%28physics%29)

- [9] Kalman filter, Wikipedia [http://en.wikipedia.org/wiki/Kalman\\_filter](http://en.wikipedia.org/wiki/Kalman_filter)
- [10] Dirac delta function, Wikipedia [http://en.wikipedia.org/wiki/Dirac\\_delta\\_function](http://en.wikipedia.org/wiki/Dirac_delta_function)
- [11] Swing-wing, Wikipedia <http://en.wikipedia.org/wiki/Swing-wing>
- [12] Moment of inertia, Wikipedia [http://en.wikipedia.org/wiki/Moment\\_of\\_inertia](http://en.wikipedia.org/wiki/Moment_of_inertia)
- [13] Astronomical instruments, Wikimedia <http://commons.wikimedia.org/wiki/Category:Astronomical>
- [14] Parallax, Wikipedia <http://en.wikipedia.org/wiki/Parallax>
- [15] Equatorial coordinate system, Wikipedia  
[http://en.wikipedia.org/wiki/Equatorial\\_coordinate\\_system](http://en.wikipedia.org/wiki/Equatorial_coordinate_system)
- [16] The Naval Observatory Merged Astrometric Dataset (NOMAD)  
[http://www.nofs.navy.mil/data/fchpix/nomad\\_format.html](http://www.nofs.navy.mil/data/fchpix/nomad_format.html)